

Important Copyright Notice:

The provision of this paper in an electronic form in this site is only for scholarly study purposes and any other use of this material is prohibited. What appears here is a near-publication draft of the final paper as appeared in the journal or conference proceedings. This is subject to the copyrights of the publishers. Please observe their copyrights.

Improved On-Line Pose Estimation for Mobile Robots by Fusion of Odometry Information and Environment Map

Behzad Moshiri
Associate Professor
Dept. of Elec. & Comp. Eng.
Faculty of Engineering
Univ. of Tehran, IRAN
moshiri@ut.ac.ir

Mohammad Reza Asharif
Professor
Dept. of Information Eng.
Faculty of Engineering
Univ. of the Ryukyus
Okinawa, Japan
asharif@ie.u-ryukyu.ac.jp

Reza HoseinNezhad
Ph.D. Student
Dept. of Elec. & Comp. Eng.
Faculty of Engineering
Univ. of Tehran, IRAN
rhnezhad@ut.ac.ir

Abstract – *A new method for on-line pose estimation in mobile robotics applications is presented in this paper. Using the information provided by the wheel encoders usually is accompanied with additive systematic and non-systematic errors. Much effort has been done to overcome this problem. In the proposed approach, it is assumed that the robot is making an occupancy grids map of the environment during exploration. Some parameters are defined for compensation of the non-systematic error, existing in the estimated value by odometry information. These parameters will be gradually estimated and tuned by a maximum likelihood estimation method. In this estimation the similarity between the current map of the environment and the local map, generated by the sensors, is considered as a likelihood ratio. Some experimental results by Khepera are presented. The results show that the performance of our method increases even compared to the case where the wheels parameters are improved by the famous UMB benchmark.*

Keywords: pose estimation, maximum likelihood, map building, occupancy grids

1 Introduction

Almost, sufficient knowledge about the robot's pose (consisting of its position and the direction angle of its motion) is essential in every mobile robotic application. Usually the wheels of the robot are equipped with some motion encoders and a pose estimation is achieved by the data, provided by these encoders. But this estimation requires some calibration because there are several sources of errors in the robot's pose, estimated by wheel encoders. For example, wear and tear can change the diameter of

wheels or loosen belts may cause odometry error and so on. Such effects can introduce significant systematic errors into the robot's odometry. The requirement for such calibration is as old as the field of robotics itself and the literature of methods for calibrating robots (for example see [4, 12]). Well-nigh, all of the exiting calibration methods have certain disadvantages. Many existing calibration methods call for human intercession i.e. in order to calibrate a mobile robot's odometry, a person (or some external agent or device) has to measure the exact position and direction angle of the robot, and deduce the physical model from these measurements. Because of two reasons, such approaches are unsuitable. Firstly a considerable amount of endeavor is involved in calibration process of the robot that normally cuts it off from its continuous operation. Secondly and more importantly, the physical characteristics of the mobile robot and the environment around it changes. For robot arms, robotic platforms and many other stationary devices, the environment is mainly static. In addition, the odometry error of the robot arm joint is strictly internal to the joint and is not deviated due to most changes of the environment.

On the other side, a mobile robot's odometry depends on the kind of surface that the robot is traveling on it. If the surface varies (e.g. from carpet to tile), then the calibration parameters change. Therefore, calibration process for such a robot, must be adaptive and in addition to pose estimation by dead-reckoning, mobile robot must fuse odometry data with a feedback from the environment. Normally, such a feedback is generated by sensory data. Existence of a rich chronicle of using multi sensor data fusion methods in mobile robotic applications (see [1] as some examples) entuses us to fuse the sen-

sory data with odometry data in order to compensate the errors and predict parameter variations. In such a localization process, robot's pose is estimated by utilizing both uncalibrated odometry and sensory data e.g. from a laser range finder sensor. Thus, the demand for a model of the odometry error is extinguished. Thrun, has done a wide research in this area [10, 11]. In his recent work [12], he implemented his algorithm on a mobile robot, that was equipped with long range finder sensors. He used a mathematical sensor model in his algorithm, that was based on ray tracing. In this method, the likelihood of a "hit" depends on the occupancy probability of the grid cell that is being traced.

In our map building and navigation experiments, we used a simple Khepera robot. This robot is equipped only with infra red proximity detectors [7]. The data, provided with such sensors, are appropriate for short distances only. Besides, there is no exact model to convert the proximity values to distance values and applying the ray tracing approach. In this paper, a new approach is introduced for compensation of the systematic part of the error that exists in odometry data. In our approach, a new maximum likelihood method is applied to estimate the parameters. In the process of approximating the likelihood ratio, the local maps that are extracted from a previously created global map, are compared with the local map that is developed by the current values of the infra red sensors of the robot. Actually, the extracted local map, depends on both the global map data and the estimation of the robot's pose. Thus, corresponding to each dead reckoning parameter values, there is a different extracted map. In our method, we seek for the parameter values, corresponding to an extracted local map, that is best fitted with the local map, generated from current sensory data.

In the second section, probabilistic formulation of pose estimation by dead reckoning (using odometry data) is briefly reviewed. Then, the new method for maximum likelihood estimation, is introduced in the third section. Experimental results will be given in the fourth section. Finally, we will give conclusions in the last section.

2 Odometry Parameters: A Brief Review

Robot motion is probabilistically modeled in our calibration method. More distinctly, let $\pi = (x, y, \theta)^T$ denote the robot's pose in a two dimensional space (θ is the robot's heading direction). Robot motion is modeled by the conditional probability distribution $P(\pi' | \pi, d)$, where π is the robot's pose before executing an action, d is the displacement measured by the robot's odometry, and π' is the pose after executing the action.

Assume that $d_{trans}(k)$ and $d_{rot}(k)$ are the estimated values of translational and rotational displacements in k^{th}

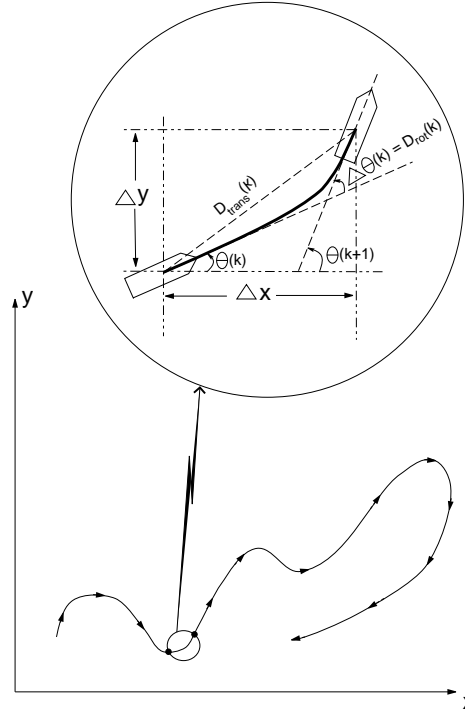


Figure 1: Robot's kinematics, based on the values of rotational and translational displacements, estimated by dead reckoning.

iteration and $D_{trans}(k)$ and $D_{rot}(k)$ are their real values respectively. If $|l|$ is the distance, traveled between the two iterations, then our probabilistic model will be expressed as below:

$$D_{trans}(k) = d_{trans}(k) + \alpha_{trans} \times |l| + e_{trans} \quad (1)$$

$$D_{rot}(k) = d_{rot}(k) + \alpha_{rot} \times |l| + e_{rot}$$

where the terms $\alpha_{trans} \times |l|$ and $\alpha_{rot} \times |l|$ stand for the *systematic* error and e_{trans} and e_{rot} are zero-mean random variables and stand for the *non-systematic* error (refer to [4] for more details on systematic and non-systematic errors of odometry data). Calibration of the robot's pose estimation is defined by estimating α_{trans} and α_{rot} .

As (1) indicates, our model presumes that both errors increase linearly with the distance traveled. In practice, it has been found that this model is superior over other various choices, including models with more parameters [12]. Figure 1 shows a typical robot's kinematics. As it is demonstrated in the figure, robot's pose state transition can be expressed by the following equation:

$$\begin{cases} \theta(k+1) = \theta(k) + D_{rot}(k) \\ x(k+1) = x(k) + D_{trans}(k) \times \cos(\theta(k) + D_{rot}(k)) \\ y(k+1) = y(k) + D_{trans}(k) \times \sin(\theta(k) + D_{rot}(k)) \end{cases} \quad (2)$$

3 Parameter Estimation

We estimate the parameters α_{trans} and α_{rot} by using the sensory data and odometry data that have been gathered during robot motion, gradually. Actually, it is a maximum likelihood estimation problem and we search for the parameters that appear most plausible under the existing sensory and odometry data. This estimation is formulated as below:

$$(\alpha_{trans}^*, \alpha_{rot}^*)^T = \operatorname{argmax} P(\alpha_{trans}, \alpha_{rot} | Q_k, S_{k+1}) \quad (3)$$

where S_{k+1} means the sensory data (e.g. infra red proximity or laser or ultrasonic range finder sensors or any other source of information related to environment perception) in the $k+1^{th}$ iteration and Q_k is the collection of the whole odometry and sensory information that is gathered in k^{th} iteration or was stored before. More specifically, it can be defined by:

$$Q_k = \{S_1, O_1, S_2, O_2, \dots, S_k, O_k\}$$

where O_i is the odometry data that is/was gathered in i^{th} iteration i.e. the erroneous measured values of translational and rotational displacements $d_{trans}(i)$ and $d_{rot}(i)$.

If the data set is large, then the maximum likelihood estimation problem is mathematically intractable (see [12]). We have made the whole algorithm simpler and adaptive. In order to consider the variations of parameters and contribute adaptation to the estimation algorithm, parameter values are estimated in each iteration by a local maximum likelihood estimator as below:

$$(\alpha_{trans}^{*(i)}, \alpha_{rot}^{*(i)})^T = \operatorname{argmax} P(\alpha_{trans}, \alpha_{rot} | Q_i, S_{i+1}) \quad (4)$$

and then, they are adapted by the following rule:

$$\begin{pmatrix} \alpha_{trans}^* \\ \alpha_{rot}^* \end{pmatrix} \leftarrow \gamma \times \begin{pmatrix} \alpha_{trans}^{*(i)} \\ \alpha_{rot}^{*(i)} \end{pmatrix} + (1-\gamma) \times \begin{pmatrix} \alpha_{trans}^{*(i)} \\ \alpha_{rot}^{*(i)} \end{pmatrix} \quad (5)$$

Here, $\gamma \leq 1$ is an exponential forgetting factor, that decays the weight of measurements over time. It is usually selected near to 1 and was 0.9 in our experiments.

3.1 Likelihood Function

The only thing that is remained to be computed or approximated, is the parameter likelihood function $P(\alpha_{trans}, \alpha_{rot} | Q_i, S_{i+1})$ in (4). According to Bayesian rule, this value can be expressed as follows:

$$P(\alpha_{trans}, \alpha_{rot} | Q_i, S_{i+1}) = \zeta \times P(S_{i+1} | Q_i, \alpha_{trans}, \alpha_{rot}) \times P(\alpha_{trans}, \alpha_{rot} | Q_i) \quad (6)$$

where $\zeta = [P(S_{i+1} | Q_i)]^{-1}$ is a normalizing factor and can be ignored during the maximization process.

The knowledge in Q_i without knowing about S_{i+1} , contains no information related to $\alpha_{trans}^{*(i)}$ and $\alpha_{rot}^{*(i)}$. So, $P(\alpha_{trans}, \alpha_{rot} | Q_i) = P(\alpha_{trans}, \alpha_{rot})$ and this is a priori probability that may also be discarded while maximization. It remains to calculate or approximate the term $P(S_{i+1} | Q_i, \alpha_{trans}, \alpha_{rot})$. We call this term *sensation likelihood*.

3.2 Estimation of the Sensation Likelihood

Assume that W is the world and $\Delta\pi$ is the relative displacement between the robot's poses π_{i+1} and π_i . According to the theorem of total probability, sensation likelihood can be calculated by:

$$P(S_{i+1} | Q_i, \alpha_{trans}, \alpha_{rot}) = \int \int P(S_{i+1} | W, \Delta\pi, Q_i, \alpha_{trans}, \alpha_{rot}) \times P(W, \Delta\pi | Q_i, \alpha_{trans}, \alpha_{rot}) dW d\Delta\pi \quad (7)$$

Since the sensor data S_{i+1} is independent from Q_i and $\alpha_{trans}, \alpha_{rot}$, and the world W and the displacement data $\Delta\pi$ are independent from each other, (7) can be expressed as below:

$$P(S_{i+1} | Q_i, \alpha_{trans}, \alpha_{rot}) = \int \int P(S_{i+1} | W, \Delta\pi) \times P(W | Q_i, \alpha_{trans}, \alpha_{rot}) \times P(\Delta\pi | Q_i, \alpha_{trans}, \alpha_{rot}) dW d\Delta\pi \quad (8)$$

Besides, we know that W is independent of the motion data and parameters. Also $\Delta\pi$ depends only on the recent motion data O_i and the α parameters. Thus (8) can be more simplified as follows:

$$P(S_{i+1} | Q_i, \alpha_{trans}, \alpha_{rot}) = \int \int P(S_{i+1} | W, \Delta\pi) \times P(W | S_1, S_2, \dots, S_i) \times P(\Delta\pi | O_i, \alpha_{trans}, \alpha_{rot}) dW d\Delta\pi \quad (9)$$

Ofcourse, integrating over all possible worlds W and all displacements $\Delta\pi$ is infeasible. We can approximate the sensation likelihood by substituting the integrals in (9) with their expected values. They are much easier to calculate. Finally the following expression, is proposed as a close approximation for the sensation likelihood:

$$P(S_{i+1} | Q_i, \alpha_{trans}, \alpha_{rot}) \cong P(S_{i+1} | W = E[W | S_1, S_2, \dots, S_i] \wedge \Delta\pi = E[\Delta\pi | O_i, \alpha_{trans}, \alpha_{rot}]) \quad (10)$$

where $E[.]$ is an indication of the conditional expected quantity of a random variable.

3.3 Calculation of the expectations and the approximated likelihood

In this subsection, it is explained that how the two expected values and the approximated likelihood in (10) are calculated in our approach. The first term,

$E[W|S_1, S_2, \dots, S_i]$ is considered as a global occupancy grids map of the environment, that has been generated up to the i^{th} iteration. There is a large set of methods for creation of such a map. It can be generated by Bayesian fusion of sensory data [5, 6] or by a more intelligent neural-Bayesian approach [13] or by our new method, called *pseudo information fusion* [2, 3]. Although there are several approaches for environment mapping in mobile robotics, occupancy grids has been selected (See [10] for more information about mapping methods and map learning). That is because a grid-based map can be easily created, handled and applied to the likelihood estimation, as it will be shown later in this subsection. The second expectation term, $E[\Delta\pi|O_i, \alpha_{trans}, \alpha_{rot}]$ is an expected displacement vector, denoted by $E[(\Delta x, \Delta y, \Delta\theta)^T|O_i, \alpha_{trans}, \alpha_{rot}]$ where $O_i = (d_{trans}(i), d_{rot}(i), |l|)^T$ is the recent movement information that is obtained by using the wheel encoders data. Equations (1) and (2) can be utilized to calculate the expected value of $\Delta\pi$, knowing the above information. But the non-systematic error terms in (1) can be ignored because during expectation calculation, their mean value is considered and they are zero.

The final term that is remained to be calculated, is the sensation likelihood itself, i.e. $P(S_{i+1}|W; \Delta\pi)$. This is the likelihood of the scan, recorded in the final position. If the sensors are long distance range finders (e.g. laser or ultrasonic sensors), then the likelihood value may be obtained by a simple ray tracing. In that case, the likelihood of a “hit” depends on the occupancy of the grid cell that is being traced. Consequently, sensor measurements that are more fitted to the occupancy grids map, will get a higher likelihood value, while measurements that contradict the map, will be assigned a lower likelihood. See [12] for more details.

If the robot is not equipped with long range finder sensors, but only with some proximity range detectors (e.g. infra red proximity detectors in the case of a Khepera miniature robot), then ray tracing can not be applied to the likelihood estimation problem. In such a case, we may use the idea of *local map matching*. It is based on the fact that any sensor scan is uniquely corresponded to a local map around the robot. More specifically, there is a transformation \mathbf{T} that transform a sensor scan S_{i+1} to a local map Γ , i.e. $\Gamma = \mathbf{T}(S_{i+1})$. In our experiments with Khepera, this transformation has been implemented by using a feed forward neural network (multi-layered perceptron). We will discuss it in the next section. It is assumed that this transformation is a one-to-one correspondence (i.e. two different sensor measurements are transformed to two different local maps around the robot). This assumption is practically valid. So, the sensation likelihood may be expressed as below:

$$P(S_{i+1}|W, \Delta\pi) = K \times P(\Gamma|W, \Delta\pi) \quad (11)$$

where K is a factor that can be ignored during maximization. Knowing a global map of the environment (or the world model W) and an estimated location of the robot in this map (calculated by the information in $\Delta\pi$), one can easily extract a previously-known local map around the robot called Γ' . We interpret the conditional probability $P(\Gamma|W, \Delta\pi)$ as a judgment about the existing fitness between these two local maps. This fitness is formulated by the following equation:

$$f(\Gamma, \Gamma') = 1 - \frac{\|\Gamma - \Gamma'\|}{\|\Gamma - \frac{1}{2}I\| + \|\Gamma' - \frac{1}{2}I\|} \quad (12)$$

where I is a matrix of the same size as Γ and Γ' with all elements equal to 1 and $\|\cdot\|$ is a matrix norm, defined by:

$$\|A\| = \sum_i \sum_j |a_{ij}| \quad (13)$$

This fitness measure, is normalized to $[0, 1]$ interval. In an ideal case, where Γ and Γ' are absolutely matched, this factor becomes 1. In the worst case, where the two maps are contradicting completely i.e. $\forall i, j; \gamma_{ij} + \gamma'_{ij} = 1$, the fraction part of the expression will have a numerator equal to its denominator and so, this factor will be zero.

Actually, only if there is sufficient occupied area around the robot, then (12) is a valid measure for the fitness between the two local maps. A threshold, μ_{min} , is introduced for the total existing occupancy in both of the local maps. Parameter calibration by maximum likelihood will take place if the following occupancy condition is true:

$$\|\Gamma\| + \|\Gamma'\| \geq \mu_{min} \quad (14)$$

As a simple example, two typical local maps around the robot are shown in figures 2 and 3. They have been extracted from a global map, corresponding with the same location of the robot while it acquires a sensor scan. The difference is that no calibration has been applied to pose estimation in figure 2 while utilization of the adapted values of α_{trans} and α_{rot} in calibration of dead reckoning in (1) has led to some rotational and translational displacements ΔL and $\Delta\theta$ in the extracted local map in figure 3. By the way, any of the two maps, can play the role of Γ' in (12) and be compared to the local map, generated from the recent sensor scan.

3.4 Step by step algorithm

Figure 4 demonstrates a step by step algorithm of our calibrated pose estimation method while the robot is exploring and mapping the environment around itself. It acquires a sensor scan S_i at the i^{th} iteration. By using the recently obtained sensory data it updates a global occupancy grids map, that has been gradually created by employing a map building method e.g. Bayesian method [5, 6], Dempster-Shafer reasoning [8] method or pseudo

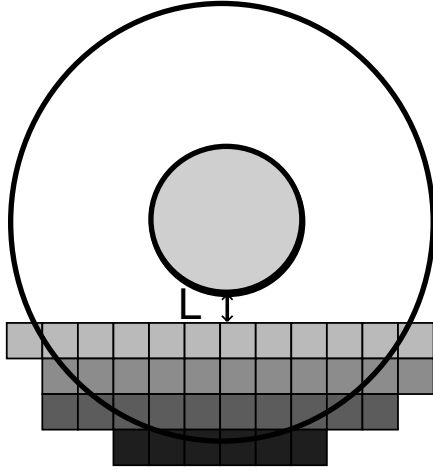


Figure 2: A local map extracted from a global occupancy grids map. No calibration has been applied to the robot's pose estimation process in this case.

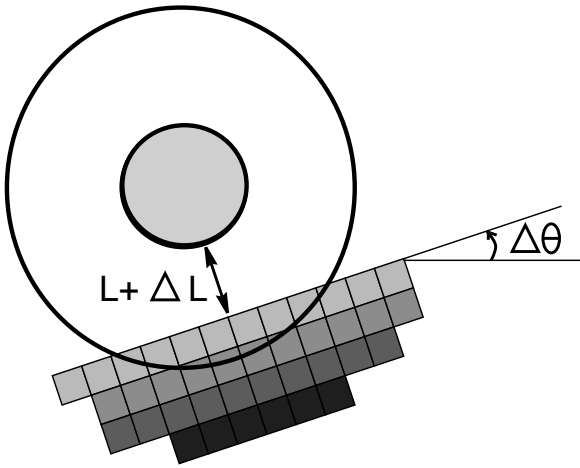


Figure 3: A local map extracted from a global occupancy grids map. Calibration of pose estimation, has led to some rotational and translational displacement in the extracted map.

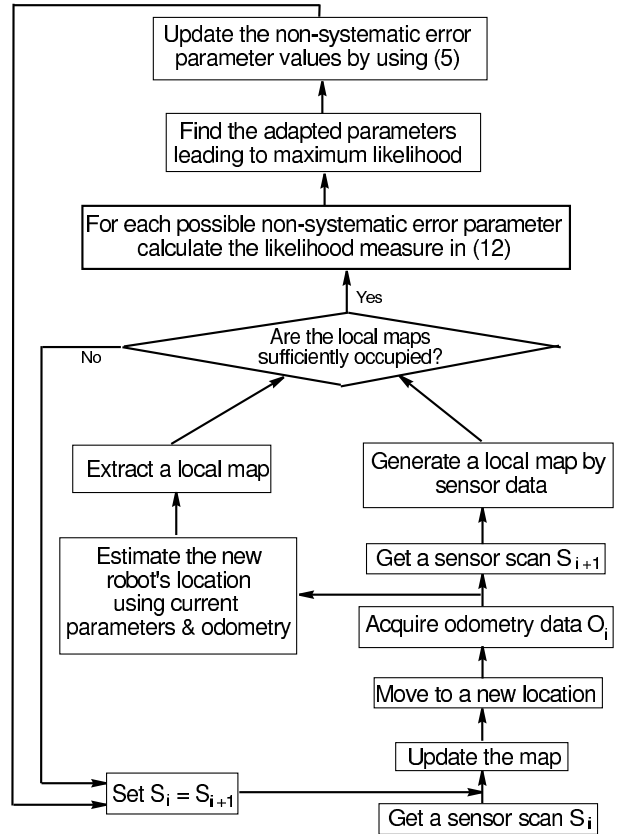


Figure 4: Our proposed step by step algorithm of calibrated pose estimation for a mobile robot, while it is exploring and mapping the environment around itself.

information method [2, 3]. The robot moves to a new location in the next step ($i + 1^{th}$ iteration). It acquires the odometry data O_i and gets a new sensor scan S_{i+1} . The robot is able to create a local map, merely by the new sensory data (without requiring any positional information). On the other hand, it estimates its new location, by making use of the odometry data and calibrating its estimate by the current dead reckoning parameters α_{trans} and α_{rot} . It uses equation (1) without the non-systematic error terms and equation (2) in this step. Now a local map can be extracted from the occupancy grids that is centered at the estimated position of the robot. This map is calculated in the same size of the local map that is generated from sensory data. In the next step, the two maps are checked if they satisfy the sufficient occupancy criterion. If they do not, algorithm skips to the next iteration by setting $S_i = S_{i+1}$ and updating the map etc. Otherwise, for each possible $(\alpha_{trans}^{*(i)}, \alpha_{rot}^{*(i)})^T$ values, the new robot's pose is re-estimated and the corresponding local map is re-extracted and the fitness measure $f(\Gamma, \Gamma')$ is calculated by (12). The parameter values which lead to maximum fitness and hence to maximum likelihood, are chosen as

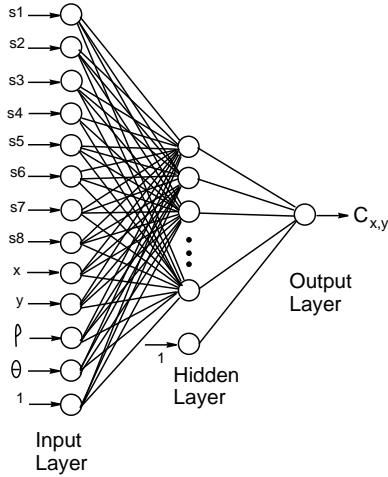


Figure 5: The feed-forward perceptron that was trained for local map building, by using proximity data, provided by Khepera’s infrared sensors.

$\alpha_{trans}^{*(i)}, \alpha_{rot}^{*(i)}$ and applied to equation (5) to adapt α_{trans} and α_{rot} values in the next step. Finally, the next iteration is started by replacing S_i with S_{i+1} and updating the map etc.

4 Experiments

Khepera miniature mobile robot has been applied to exploration, localization and map building experiments. In these experiments, the data provided by 8 infra-red sensors around Khepera, were the only sources of information (See [7] for more details on Khepera sensors and structure). There is no accurate inverse model for the infrared proximity detectors, like the model for ultrasonic range finders. We trained a feed-forward multi-layered perceptron to implement an inverse model for the sensors. The inputs of the network are the eight proximity values, and the local coordinates of a cell in the occupancy grids map around the robot. The output of the network is the occupancy probability value of the cell. The architecture of the neural network is shown in figure 5. This output value is fused with the associated occupancy probability value of the same cell in a global map of the environment. This global map has been calculated based upon previous sensory information gradually. Finally the resulting probability is applied to improve the global map. In other words, during exploration of the environment by the reactive obstacle avoidance method of Braitenberg, Khepera creates a local occupancy grids map in every sensing iteration. These local maps are integrated with an initially blank global map and it improves gradually. Another more intelligent alternative for obstacle avoidance is our newly introduced method of fuzzy rule-based command fusion [9]. Figure 6, shows the typical environment in which our

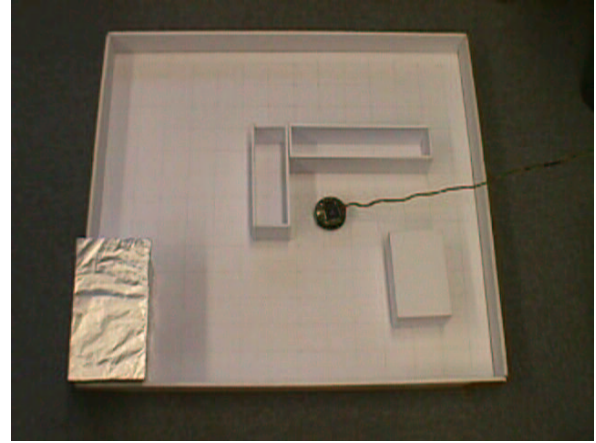


Figure 6: A photo of Khepera and the environment, in our experiments.

experiments have been done.

The well-known UMBmark (University of Michigan Benchmark) was tried in the first experiment. The robot travels a square path with a perimeter of 200cm length, 16 times in clockwise and 16 times in counterclockwise direction. Figure 7 shows the occupancy grids map, created by pseudo information fusion, and the square path inside it. Practically, the robot does not travel the exact square path in each round, because the motion control commands are created based on the robot’s pose information, acquired from the erroneous odometry data. It causes the robot not to return to its initial starting point in each round. This deviation of the location of the robot with respect to the original point of motion starting was manually measured. In the first trial of 32 rounds, their mean values were $\overline{\Delta x} = 35.5\text{mm}$ and $\overline{\Delta y} = 26.3\text{mm}$, equivalent to a relative error of 2.21%¹. The UMBmark calibration parameters were calculated to be $E_d = 0.9991$ and $E_b = 0.9953$. They are some correction factors for wheel diameter and wheel base, respectively. They were computed, based on the results that were obtained in the first trial, in order to decrease the systematic odometry error (Refer to [4] for more details on UMBmark basics and formulation). In the second trial, the calibrated values of the wheel diameter and base, were applied to odometry and motion command generation process. This calibration caused a lower average deviation of $\overline{\Delta x} = 15.3\text{mm}$ and $\overline{\Delta y} = 12.8\text{mm}$, or in other words a relative error of 0.99%. In the third trial, we applied our proposed calibration method to the pose estimation and motion commands generation. It was observed that the robot traverses the square path more accurately and the average position deviation was measured as $\overline{\Delta x} = 3.2\text{mm}$ and

¹This relative error is computed by $100\% \times \frac{\overline{\Delta x}^2 + \overline{\Delta y}^2}{4 \times L^2}$ where $L = 50\text{mm}$ is the side length of the square path.

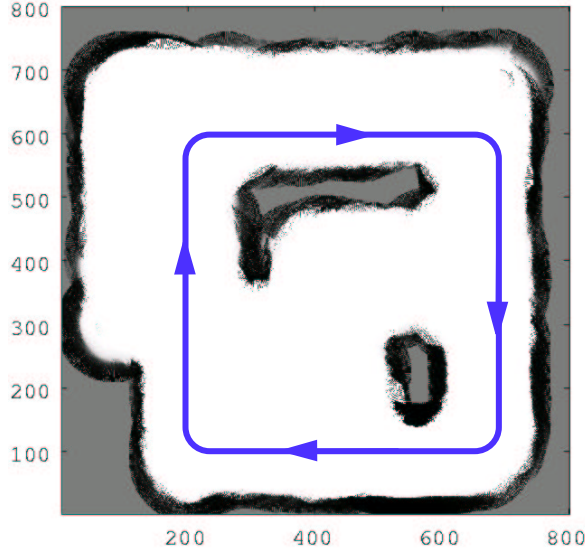


Figure 7: The square path, traveled by the robot in our UMBmark experiment.

$\overline{\Delta y} = 1.6\text{mm}$. This is equivalent to a relative error of 0.178%. It is seen that relative estimation error has decreased by 82%.

In the second experiment, the robot was moved from a start point to an end point in a specific route. Two routes are shown in figure 8. The actual path and the path that was estimated to be traversed by the robot, are depicted by solid and dash line respectively. It is observed that the estimated route deviates gradually. Finally, the end point that was actually reached by the robot is located in a far distance from the estimated end point and following errors were measured:

$$\begin{aligned} |\Delta x| &= |x_{\text{End}} - x_{\text{Estimated End}}| = 269\text{mm} \\ |\Delta y| &= |y_{\text{End}} - y_{\text{Estimated End}}| = 168\text{mm} \end{aligned}$$

At the same experiment, the same odometry data were applied to our proposed calibrated version of dead reckoning method and another estimation for the path was obtained. Both the actual and the new estimated paths are depicted in figure 9. In contrast with the previous case, it is observed that the estimated path is much closer to the real one. That is because of the efficient calibration of systematic error of odometry data by using the adapting parameters α_{trans}^* and α_{rot}^* . Indeed, end position estimation errors were measured as below:

$$\begin{aligned} |\Delta x'| &= |x_{\text{End}} - x_{\text{Estimated End}}| = 25\text{mm} \\ |\Delta y'| &= |y_{\text{End}} - y_{\text{Estimated End}}| = 38\text{mm} \end{aligned}$$

Thus, estimation error has been reduced by $\left(1 - \frac{\sqrt{|\Delta x'|^2 + |\Delta y'|^2}}{\sqrt{|\Delta x|^2 + |\Delta y|^2}}\right) \times 100\% = 85.66\%$.

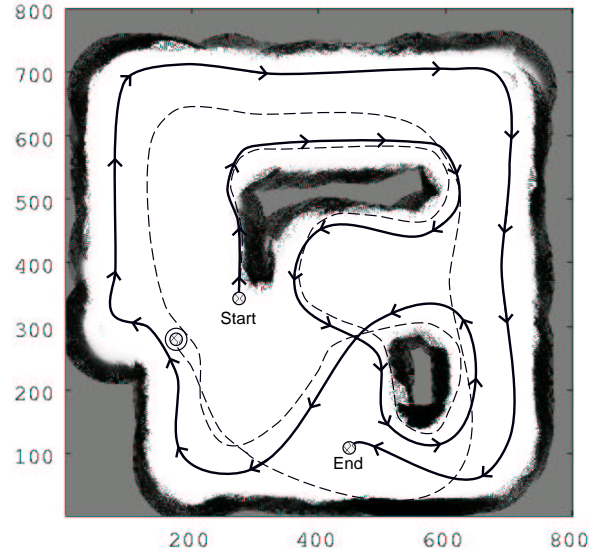


Figure 8: The real and the estimated paths, traversed by the robot are shown by solid and dash lines respectively.

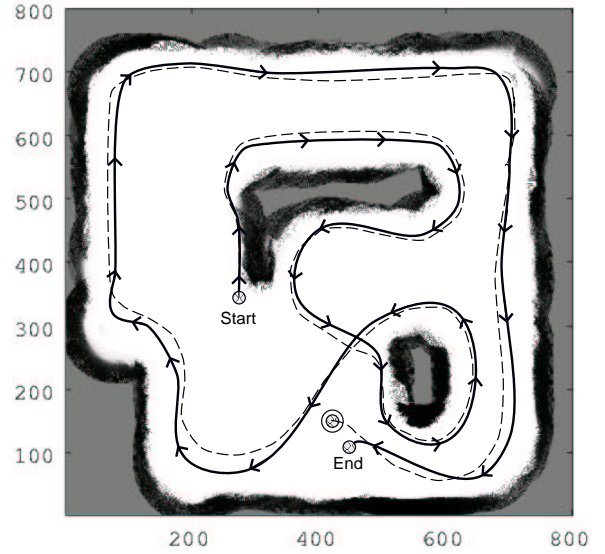


Figure 9: The real and the estimated paths, traversed by the robot are shown by solid and dash lines respectively. In this case, our proposed calibration method has been applied to the pose estimation to generate the estimated path. That is why it is much closer to the actual path.

5 Conclusions

A new approach for calibration of dead reckoning process in pose estimation of mobile robots was proposed in this paper. We attempted to model the systematic part of the error, existing in odometry data, by introducing two calibration parameters and estimating them adaptively. It was suggested to fuse the current sensory and odometry information with the information that had been obtained previously. The local map, generated by current sensory data, is integrated with a local map that is extracted from a global occupancy grids map of environment. Actually, the most desired parameters are those which lead to a mostly fitted version of an extracted local map. It was shown that the proposed approach gives a maximum likelihood estimation of the parameters. The most important advantages of the method are its simplicity and its applicability to the cases where the robot is equipped with short range distance sensors (proximity detectors). In our experiments, Khepera robot was utilized to examine the performance of the method for reduction of pose estimation error. The proximity data, provided by the infra red sensors of the robot and the odometry data, provided by wheel encoders, were the only sources of information in these experiments. Results show that our approach causes a reduction of more than 80% in pose estimation error. Besides, the proposed approach is applicable in an online case. It means that the robot may calibrate its position by this method, while it is exploring the environment around itself and mapping it. In this case, the important difference is the fact that there is no previously created global map for extraction of local maps. But it seems very probable that a partially generated map can be useful to be applied in our method.

Acknowledgement

This research work was supported through mutual research program between University of Tehran, Faculty of Engineering (IRAN) and University of the Ryukyus, Faculty of Engineering, Department of Information Engineering (JAPAN) by Monbusho scholarship scheme.

References

- [1] A. Abidi and R.C. Gonzalez: Data Fusion in Robotic and Machine Intelligence, Academic Press, 1992
- [2] M.R. Asharif and B. Moshiri and R. HoseinNezhad: Pseudo Information Measure: A New Concept for Sensor Data Fusion, Applied in Map Building for Mobile Robots, Proc. of Int. Conf. on Signal Processing Applications and Technology (ICSPAT 2000), Dallas, Texas, USA, 16-19 October 2000
- [3] M.R. Asharif and B. Moshiri and R. HoseinNezhad: Environment Mapping for Khepera Robot: A New Method by Fusion of Pseudo Information Measures, Proc. of International Symposium on Artificial Life and Robotics (AROB), Tokyo, Japan, January 2001
- [4] J. Borenstein and H. R. Everett and L. Feng: Where am I? Sensors and Methods for Mobile Robot Positioning, Technical Report, University of Michigan, <ftp://ftp.eecs.umich.edu/people/johannb/pos96rep.pdf>, 1996
- [5] A. Elfes: Using Occupancy Grids for Mobile Robot Perception and Navigation, *Computer* 22(6), 249-265, 1989
- [6] A. Elfes: Sonar-Based Real-World Mapping and Navigation, *IEEE J. of Robotics and Automation*, 249-265, 1987
- [7] K-Team S. A.: Khepera User Manual(5.0 ed.), Lausanne, Switzerland, 1998
- [8] R. Murphy: Dempster-Shafer Theory for Sensor Fusion in Autonomous Mobile Robots, *IEEE Trans. on Robotics and Automation* VOL. 14, NO.2, 1998.
- [9] B. Moshiri and A. Eydgahi and M. Najafi and R. Hosein Nezhad: Multi-Sensor Data Fusion Used in Intelligent Autonomous Navigation, IASTED - CA'99 (Control Applications), Banff, Canada, 25-29 July 1999.
- [10] S. Thrun: Learning Maps for Indoor Mobile Robot Navigation, *Artificial Intelligence*, 1998
- [11] S. Thrun: An approach to learning mobile robot navigation, *Robotics and automation systems* 15, 301-319, 1995
- [12] S. Thrun and W. Burgard and D. Fox: A Probabilistic Approach to Concurrent Mapping and Localization for mobile robots, *Machine Learning* 31, 29-53, 1998
- [13] J.W.M. Van Dam: Environment Modeling for Mobile Robot: Neural Learning for Sensor Fusion, University of Amsterdam, Ph.D. Thesis, 1998